Constructing an Installer Package

Session 106

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

Constructing an Installer Package

Deliver a Great First Impression

Jean-Pierre Ciudad
Manager
Installation and Setup Group

http://developer.apple.com/wwdc2005/

Welcome!

- Hands-On
- Packages 101
- Distribution Scripts
  - Requirement Checks
  - Controlling What Gets Installed
- Homework (sample code)
  - Multi-CD installing
  - Relocatable Installs

Creating Packages

What are packages?

- Documents used by Installer
- They contain:
  - Your software
  - Requirement checks
  - UI resources

Creating Packages

Getting started

- Divide into several packages
  - Relocatable items (usually apps)
  - Different authorization levels
  - Small granularity = more flexibility

Creating Packages

Getting started

- Bundle Identifiers are important
  - Different .pkgs need different Bundle IDs
  - Used to match packages on upgrade

Creating Packages

Getting started

- Pick default location and relocatability
  - /Applications
  - /Library/Fonts
  - “/” if your root is built from there—not relocatable

Creating Packages

PackageMaker

- We will use four packages
  - myApp.pkg—application bundle, relocatable
  - myFont.pkg—a font installed in /Library/Fonts
  - myHelp.pkg—help files installed in /Library/Application Support/myApp
  - myDocs.pkg—a set of example documents
- Use PackageMaker to create each of the four packages
  - Can also use `packagemaker` command line tool

Hands-On #1

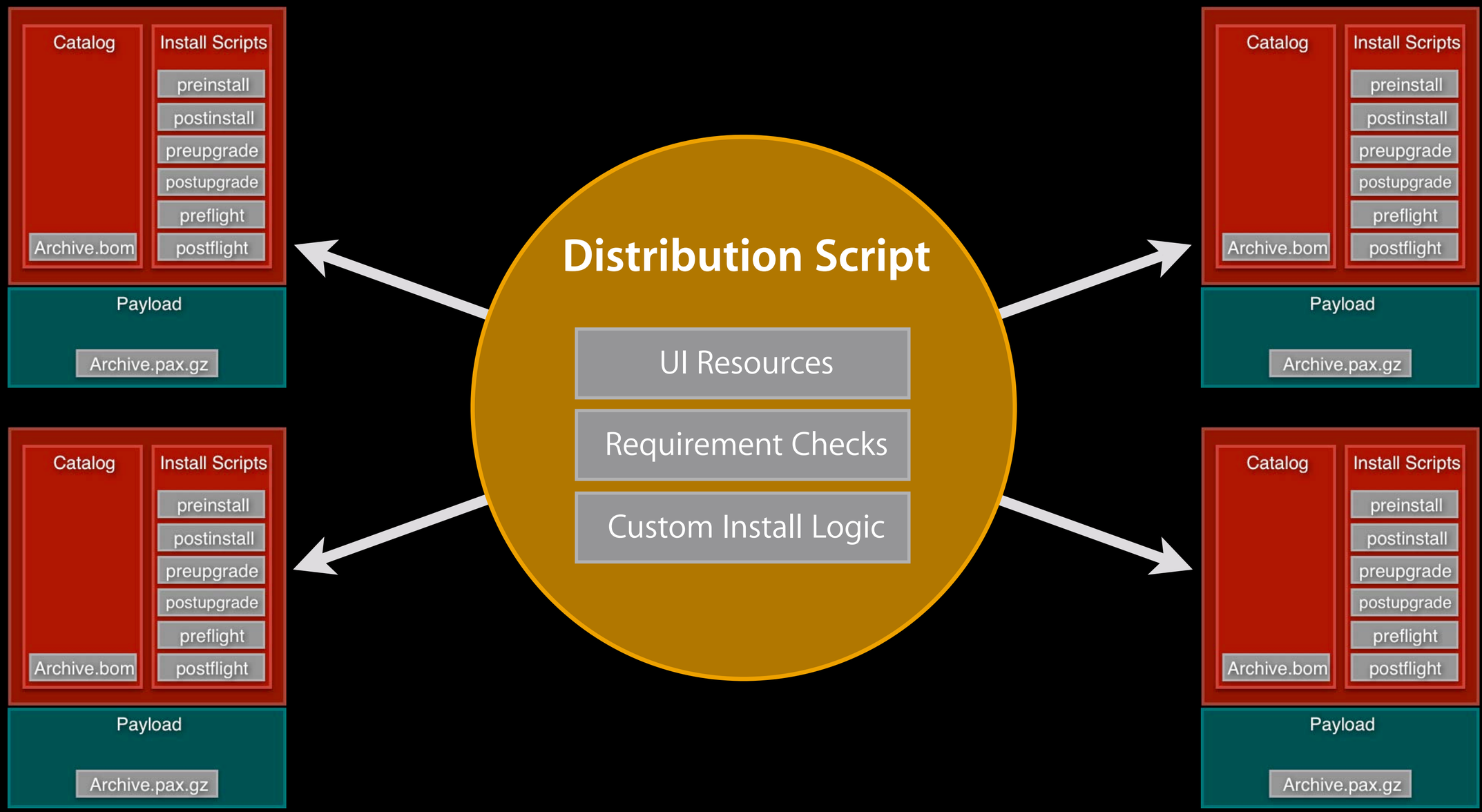
PackageMaker

http://developer.apple.com/wwdc2005/

Introduction to Distribution Scripts

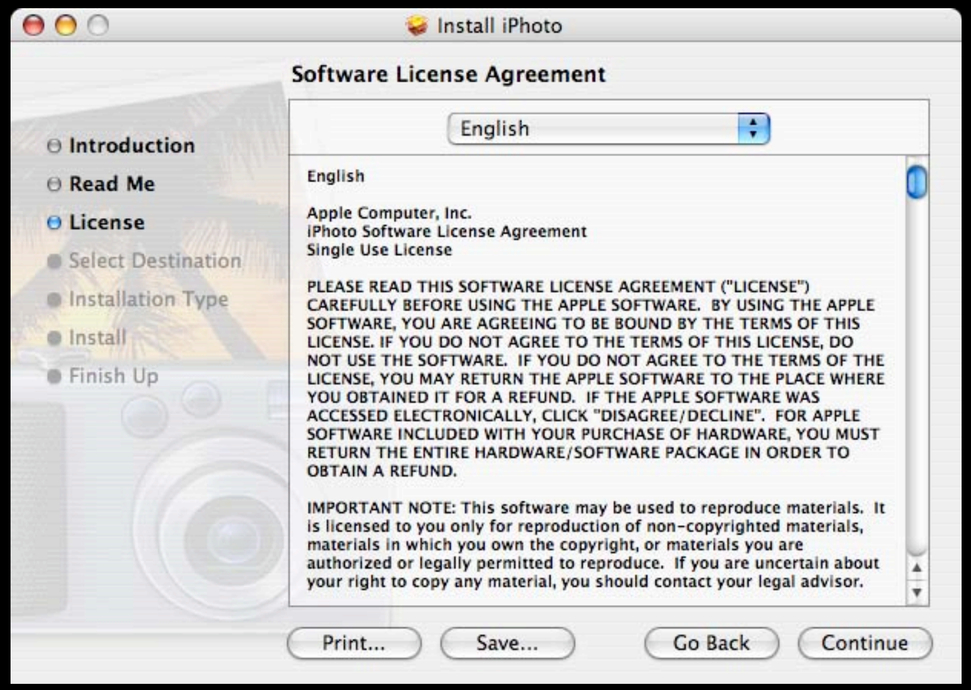
Combine Packages with Distribution Scripts

Take control of the Installer's UI in one file



UI Resources

- Document Resources
  - Welcome
  - Read Me
  - License
  - Conclusion
- Background Image
- Localizations



Document Resources

- Documents
  - Welcome, Read Me, License, and Conclusion
- XML elements
  - <welcome/>, <readme/>, <license/>, and <conclusion/>
- The “file” attribute

```
<welcome file='Welcome.rtfd'/>
<readme file='ReadMe.txt'/>
<license file='License.rtf'/>
<conclusion file='Conclusion.rtf'/>
```

Document Resources

- Documents
  - Welcome, Read Me, License, and Conclusion
- XML elements
  - <welcome/>, <readme/>, <license/>, and <conclusion/>
- The “file” attribute

```
<welcome file='Welcome.rtfd'/>
<readme file='ReadMe.txt'/>
<license file='License.rtf'/>
<conclusion file='Conclusion.rtf'/>
```

Document Resources

- Documents
  - Welcome, Read Me, License, and Conclusion
- XML elements
  - <welcome/>, <readme/>, <license/>, and <conclusion/>
- The “file” attribute

```
<welcome file='Welcome.rtfd'/>
<readme file='ReadMe.txt'/>
<license file='License.rtf'/>
<conclusion file='Conclusion.rtf'/>
```

Document Resources

- Documents
  - Welcome, Read Me, License, and Conclusion
- XML elements
  - <welcome/>, <readme/>, <license/>, and <conclusion/>
- The “file” attribute

```
<welcome file='Welcome.rtfd'/>
<readme file='ReadMe.txt'/>
<license file='License.rtf'/>
<conclusion file='Conclusion.rtf'/>
```

Document Resources

- Documents
  - Welcome, Read Me, License, and Conclusion
- XML elements
  - <welcome/>, <readme/>, <license/>, and <conclusion/>
- The “file” attribute

```
<welcome file='Welcome.rtfd'/>
<readme file='ReadMe.txt'/>
<license file='License.rtf'/>
<conclusion file='Conclusion.rtf'/>
```

Document Resources

- Documents
  - Welcome, Read Me, License, and Conclusion
- XML elements
  - <welcome/>, <readme/>, <license/>, and <conclusion/>
- The “file” attribute

```
<welcome file='Welcome.rtfd'/>
<readme file='ReadMe.txt'/>
<license file='License.rtf'/>
<conclusion file='Conclusion.rtf'/>
```

Background Image

- XML element <background/>
- “file” attribute

```
<background file='background.tiff' scaling='none' alignment='left'/>
```

Background Image

- XML element <background/>
- “file” attribute

```
<background file='background.tiff' scaling='none' alignment='left'/>
```

Background Image

- XML element <background/>
- “file” attribute
- “scaling” attribute
  - none, tofit, and proportional
  - tofit is the default

```
<background file='background.tiff' scaling='none' alignment='left'/>
```

Background Image

- XML element <background/>
- “file” attribute
- “scaling” attribute
  - none, tofit, and proportional
  - tofit is the default
- “alignment”
  - left, right, center, top, bottom, topleft, topright, bottomleft, and bottomright
  - center is the default

```
<background file='background.tiff' scaling='none' alignment='left'/>
```

Localization

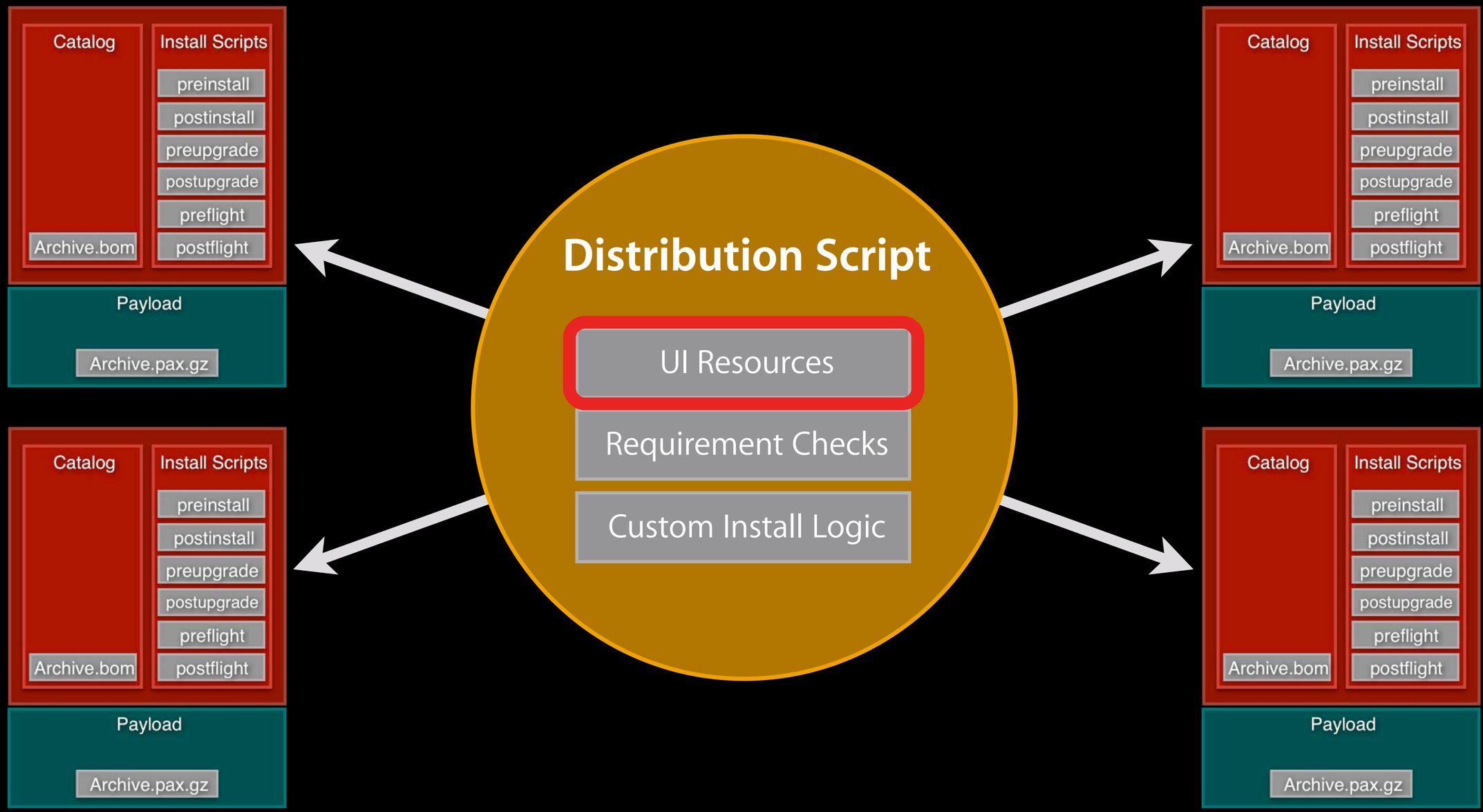
- Use an '.mpkg'
  - It's a bundle
- Enlish.lproj, Japanese.lproj, etc
- Localizable.strings
- Located in the metapackage's 'Resources' folder

Hands-On #2

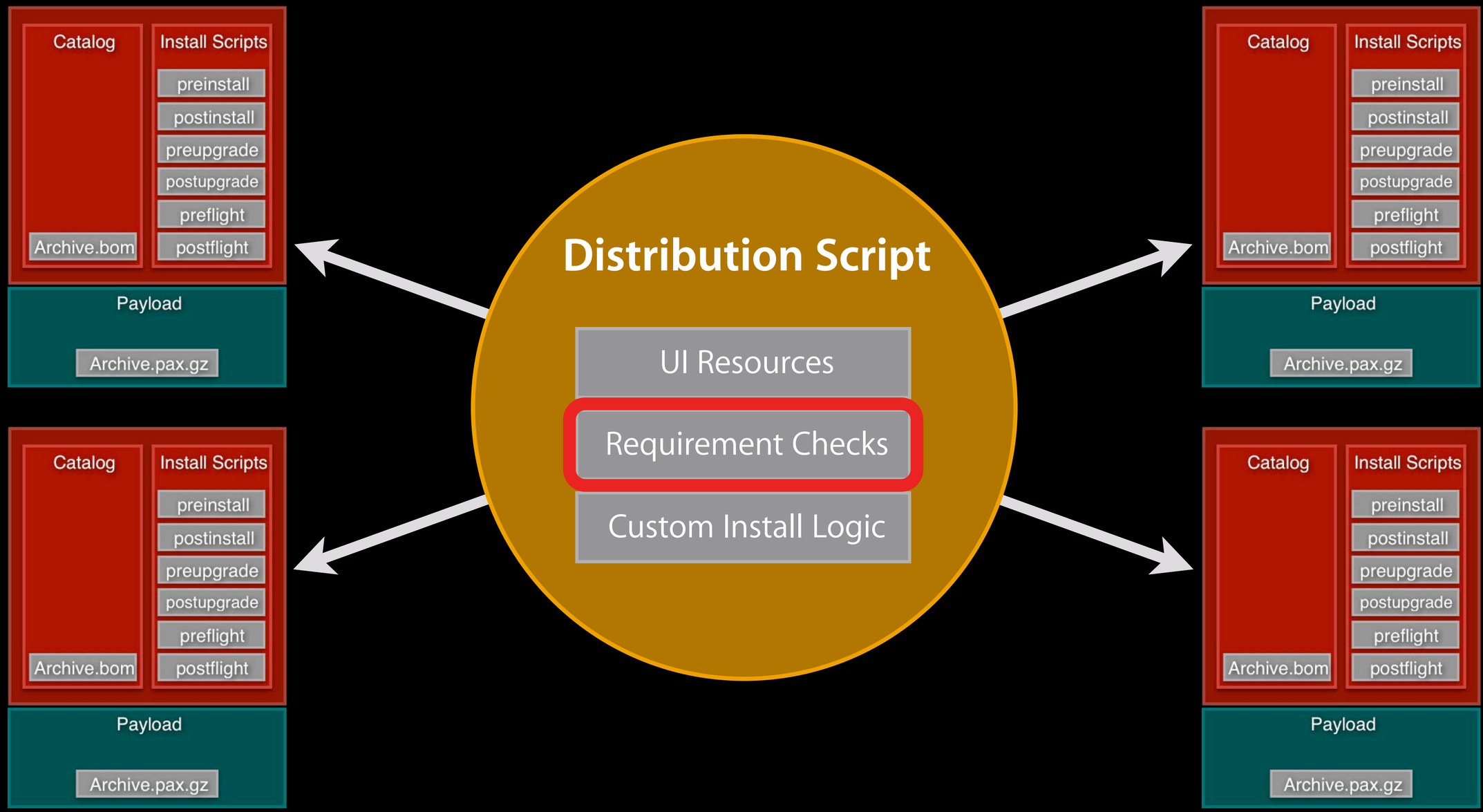
Introduction to Distribution Scripts

http://developer.apple.com/wwdc2005/

Combine Packages with Distribution Scripts



Combine Packages with Distribution Scripts



Requirement Checks

Chris Ryan
Apple Software Engineer
Installation and Setup Group

http://developer.apple.com/wwdc2005/

Requirement Checks

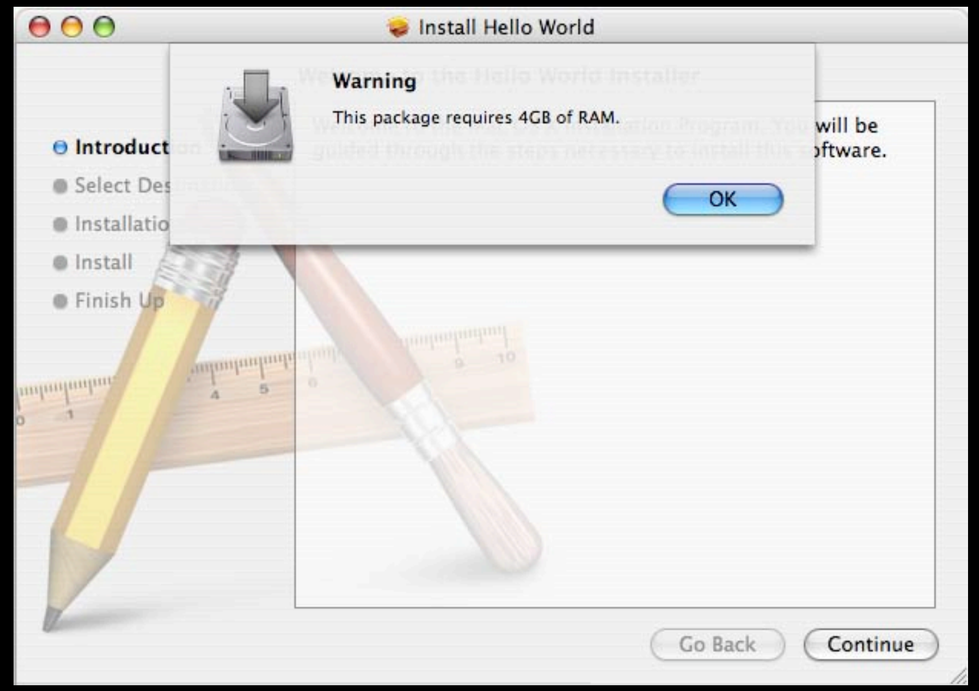
Overview

- Concepts
- How to use them in Distribution Scripts
- Demo

Requirement Checks

Concepts

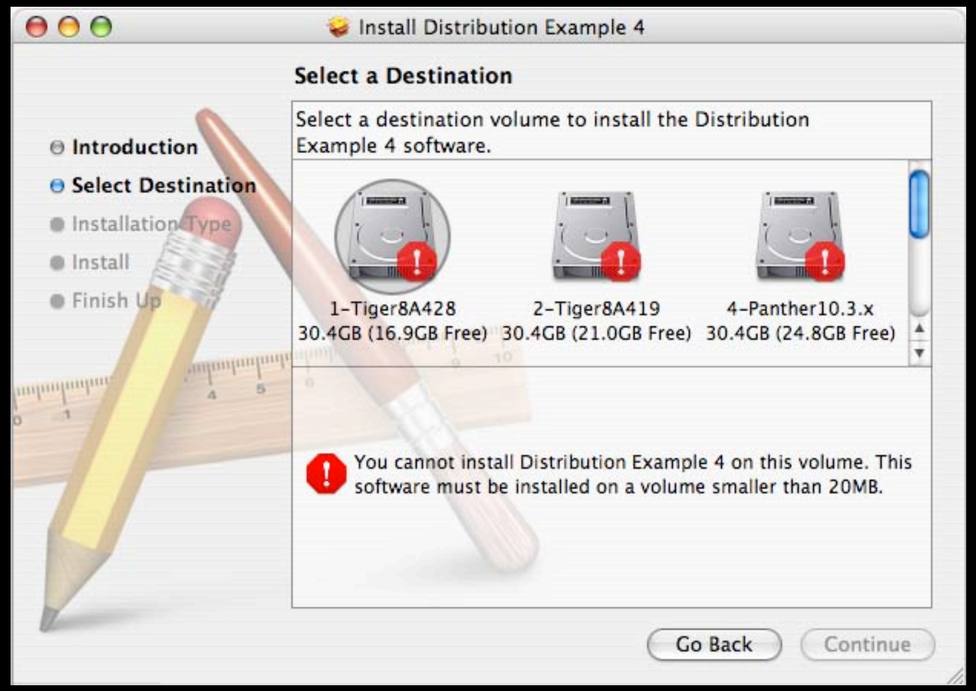
- Check Prerequisites
  - Installation Check



Requirement Checks

Concepts

- Check Prerequisites
  - Installation Check
  - Volume Check



Requirement Checks

In Distribution Scripts

- Written in JavaScript
- Read-only information about the system
- Explain requirements to the user

Installation Check

Working with JavaScript

- <installation-check/> element
  - “script” attribute
    - Contains a JavaScript snippet, usually a single function call
- <script/> element
  - Your JavaScript functions

```
<installation-check script='allow_anything()'/>

<script>
  function allow_anything() {
     return true;
  }
</script>
```

Installation Check

Working with JavaScript

- <installation-check/> element
  - "script" attribute
    - Contains a JavaScript snippet, usually a single function call
- <script/> element
  - Your JavaScript functions

```
<installation-check script='allow_anything()'/>

<script>
  function allow_anything() {
     return true;
  }
</script>
```

Installation Check

Working with JavaScript

- <installation-check/> element
  - “script” attribute
    - Contains a JavaScript snippet, usually a single function call
- <script/> element
  - Your JavaScript functions

```
<installation-check script='allow_anything()'/>

<script>
  function allow_anything() {
     return true;
  }
</script>
```

Installation Check

Working with JavaScript

- Installer's built in JavaScript global objects
  - system.*
    - system.log()
    - system.ioregistry.*
    - system.sysctl()
    - Many more…

```
<script>
   function hello_world(){
      system.log('hello world');
   }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
   function checkRAM() {
      var systemMemory = system.sysctl('hw.physmem');

      if ( systemMemory < (256*1024*1024) ) {
         my.result.type = 'Fatal';
         my.result.title = 'Requirement Failure';
         my.result.message = 'There is not enough memory to install.';
         return false;
      }
      return true;
   }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Installation Check

Returning results

```
<installation-check script='checkRAM()'/>

<script>
  function checkRAM() {
     var systemMemory = system.sysctl('hw.physmem');

     if ( systemMemory < (256*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.title = 'Requirement Failure';
        my.result.message = 'There is not enough memory to install.';
        return false;
     }
     return true;
  }
</script>
```

Volume Check

- Checks requirements against a given volume
- Runs for each volume
- Has access to information about the volume
- No warnings, just failures

Volume Check

Checking Target requirements

- “my.target”
  - my.target.mountpoint
  - my.target.availableKilobytes
  - my.target.systemVersion
  - my.target.receiptForIdentifier()
    - Returns an object with the identifier and version attributes
- “choices”
  - choices[’TextEditChoice’]
  - More on this later

Volume Check

- <volume-check/> element

```
<volume-check script='volume_has_100GB_free()'/>

<script>
  function volume_has_100GB_free() {
     if ( my.target.availableKilobytes < (100*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.message = 'MyGreatApp requires 100GB of free space.';
        return false;
     }
     return true;
  }
</script>
```

Volume Check

- <volume-check/> element

```
<volume-check script='volume_has_100GB_free()'/>

<script>
  function volume_has_100GB_free() {
     if ( my.target.availableKilobytes < (100*1024*1024) ) {
        my.result.type = 'Fatal';
        my.result.message = 'MyGreatApp requires 100GB of free space.';
        return false;
     }
     return true;
  }
</script>
```

Hands-On #3

Requirement Checks

http://developer.apple.com/wwdc2005/

Hands-On

Requirement Checks

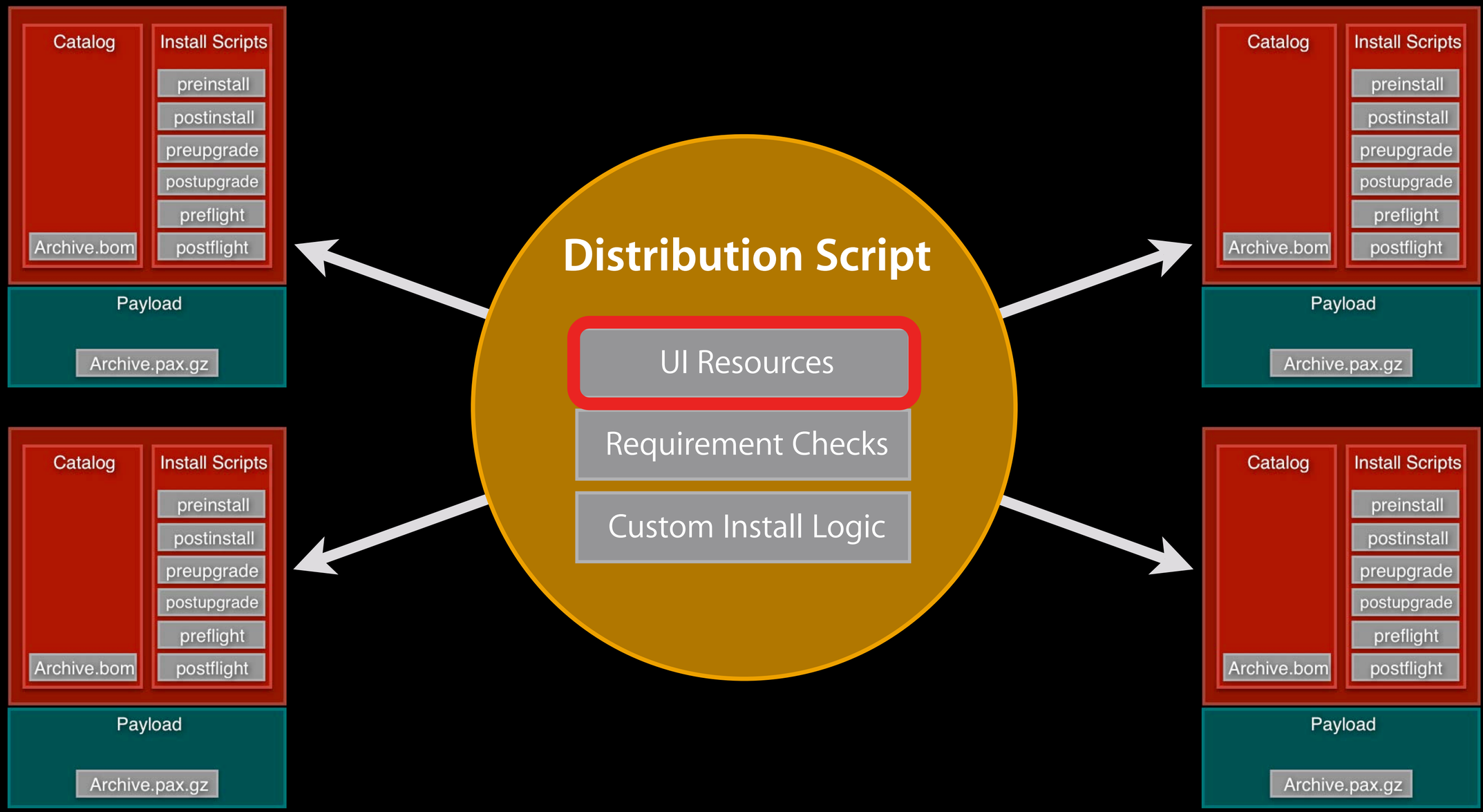
- <script/> section
- Hello World <volume-check/> from scratch
  - Hello World, this volume is "Macintosh HD"
- Convert a <installation-check/> that fails into a warning

Hands-On #3

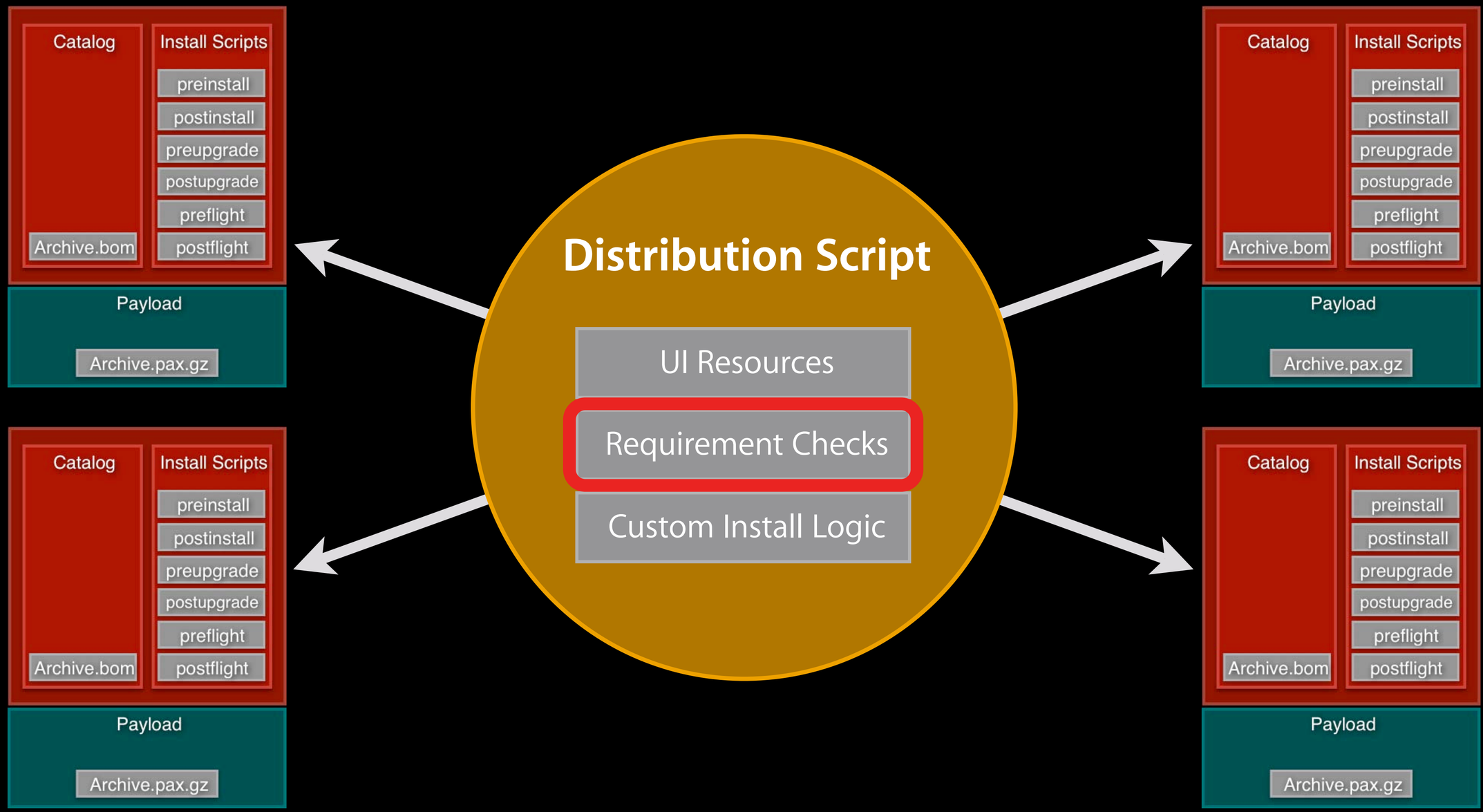
Requirement Checks

http://developer.apple.com/wwdc2005/

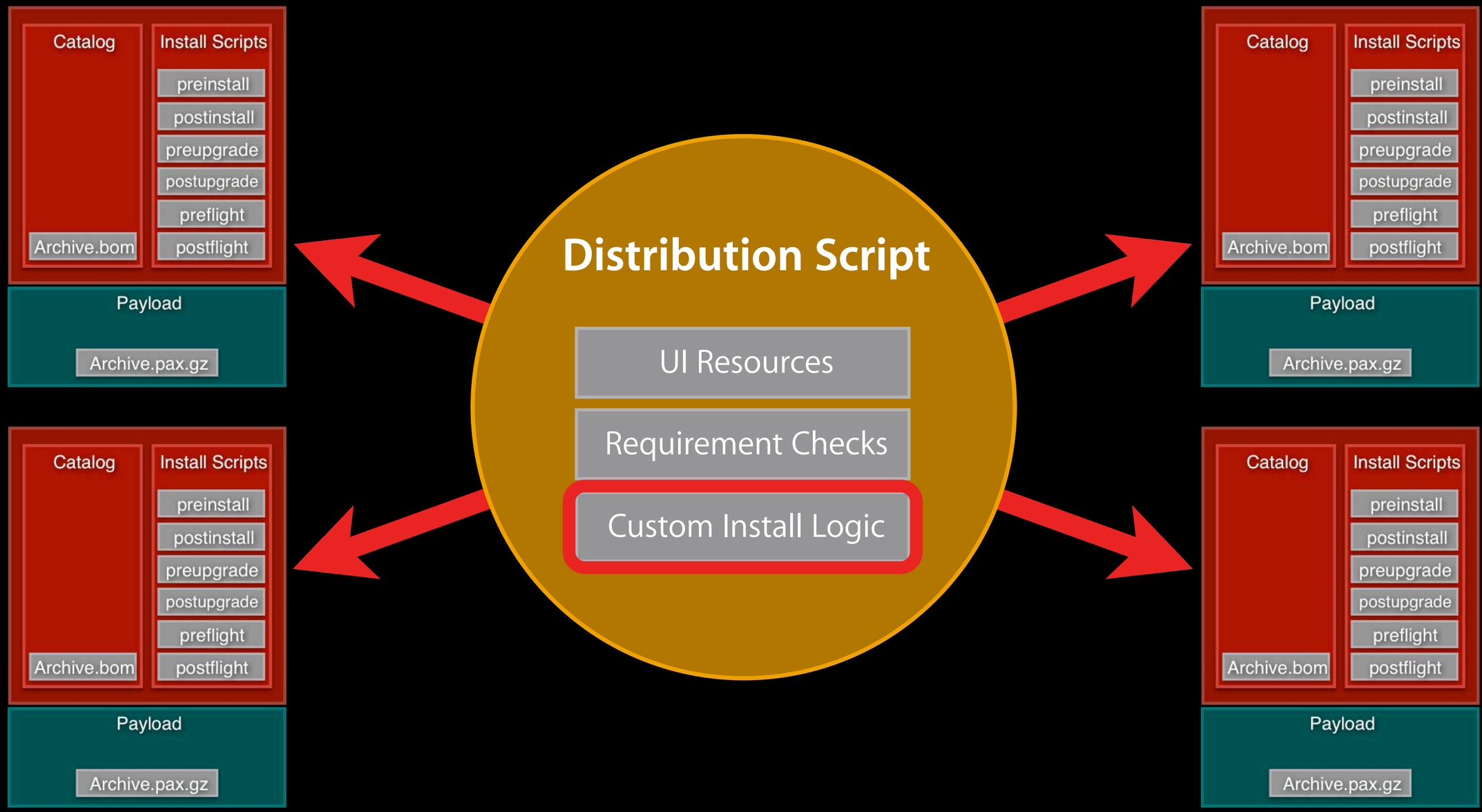
Combine Packages with Distribution Scripts



Combine Packages with Distribution Scripts



Combine Packages with Distribution Scripts



Controlling What Gets Installed

Peter Bierman
Apple Software Engineer
Installation and Setup Group

http://developer.apple.com/wwdc2005/

What You'll Learn

- Package References
- Choices
- Easy and Custom Install
- How the user customizes the install
- How to limit that customization

Package Reference

- Each package has a <pkg-ref/> element in the distribution script
  - <pkg-ref/> is the representation of a real ".pkg" package

```
<pkg-ref
  id='com.apple.wwdc.handson.myApp'
  installKBytes='344'>
  file:Contents/Packages/MyApp.pkg
</pkg-ref>
```

Package Reference

- Each package has a <pkg-ref/> element in the distribution script
  - <pkg-ref/> is the representation of a real ".pkg" package
- <pkg-ref/> elements should contain
  - An identifier "id"
    - CFBundleIdentifier
  - A size "installKBytes"
  - A file URL

```
<pkg-ref
  id='com.apple.wwdc.handson.myApp'
  installKBytes='344'>
  file:Contents/Packages/MyApp.pkg
</pkg-ref>
```

Package Reference

- Each package has a <pkg-ref/> element in the distribution script
  - <pkg-ref/> is the representation of a real ".pkg" package
- <pkg-ref/> elements should contain
  - An identifier "id"
    - CFBundleIdentifier
  - A size "installKBytes"
  - A file URL

```
<pkg-ref
  id='com.apple.wwdc.handson.myApp'
  installKBytes='344'>
  file:Contents/Packages/MyApp.pkg
</pkg-ref>
```

Package Reference

- Each package has a <pkg-ref/> element in the distribution script
  - <pkg-ref/> is the representation of a real ".pkg" package
- <pkg-ref/> elements should contain
  - An identifier "id"
    - CFBundleIdentifier
  - A size "installKBytes"
  - A file URL

```
<pkg-ref
  id='com.apple.wwdc.handson.myApp'
  installKBytes='344'>
  file:Contents/Packages/MyApp.pkg
</pkg-ref>
```

Package Reference

- Each package has a <pkg-ref/> element in the distribution script
  - <pkg-ref/> is the representation of a real ".pkg" package
- <pkg-ref/> elements should contain
  - An identifier "id"
    - CFBundleIdentifier
  - A size "installKBytes"
  - A file URL

```
<pkg-ref
  id='com.apple.wwdc.handson.myApp'
  installKBytes='344'>
  file:Contents/Packages/MyApp.pkg
</pkg-ref>
```

Packages and the Installer UI

- The <choice/> element is a line item in the Custom Install panel
  - It is the bridge between the UI and the actual ".pkg" package

```
<choice id='myApp' title='My Great App'>
  <pkg-ref id='com.apple.wwdc.handson.myApp'/>
</choice>
```

Packages and the Installer UI

- The <choice/> element is a line item in the Custom Install panel
  - It is the bridge between the UI and the actual ".pkg" package
- A <choice/> element includes:
  - An identifier "id"
  - A user-visible title "title"
  - Zero or more references to <pkg-ref/> elements

```
<choice id='myApp' title='My Great App'>
  <pkg-ref id='com.apple.wwdc.handson.myApp'/>
</choice>
```

Packages and the Installer UI

- The <choice/> element is a line item in the Custom Install panel
  - It is the bridge between the UI and the actual ".pkg" package
- A <choice/> element includes:
  - An identifier "id"
  - A user-visible title "title"
  - Zero or more references to <pkg-ref/> elements

```
<choice id='myApp' title='My Great App'>
  <pkg-ref id='com.apple.wwdc.handson.myApp'/>
</choice>
```

Packages and the Installer UI

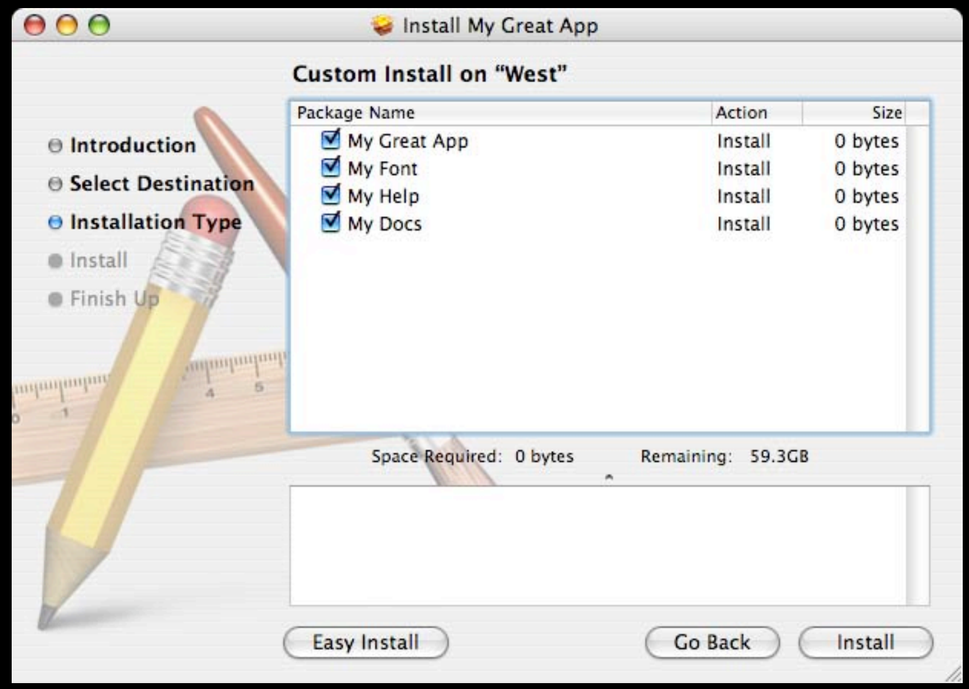
- The <choice/> element is a line item in the Custom Install panel
  - It is the bridge between the UI and the actual ".pkg" package
- A <choice/> element includes:
  - An identifier "id"
  - A user-visible title "title"
  - Zero or more references to <pkg-ref/> elements

```
<choice id='myApp' title='My Great App'>
  <pkg-ref id='com.apple.wwdc.handson.myApp'/>
</choice>
```

Custom Install Panel

- The <choices-outline/> element controls the layout of choices in the Custom Install panel

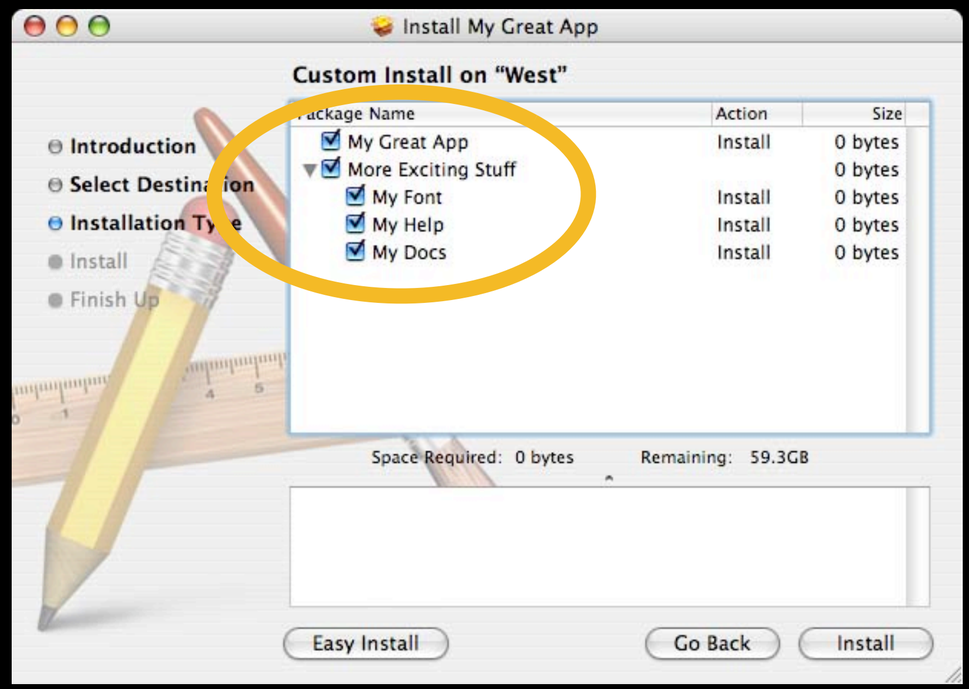
```
<choices-outline>
  <line choice='myApp'/>
  <line choice='myFont'/>
  <line choice='myHelp'/>
  <line choice='myDocs'/>
</choices-outline>
```



Custom Install Panel

- The <choices-outline/> element controls the layout of choices in the Custom Install panel
- You can nest <line/> elements to create groups

```
<choices-outline>
   <line choice='myApp'/>
   <line choice='excitingGroup'>
      <line choice='myFont'/>
      <line choice='myHelp'/>
      <line choice='myDocs'/>
   </line>
</choices-outline>
```



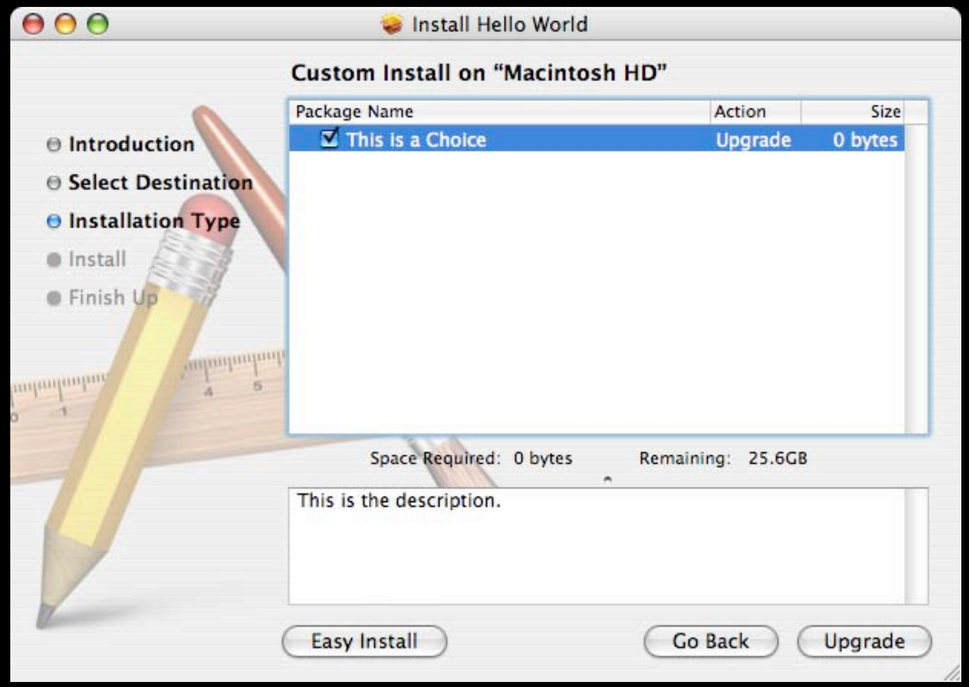
Easy and Custom Install

- Easy Install is default selections from a Custom Install
- Most users simply wish to click straight through
- If you want, you can lock in either mode using

```
<options customize='always'/>
or
<options customize='never'/>
```

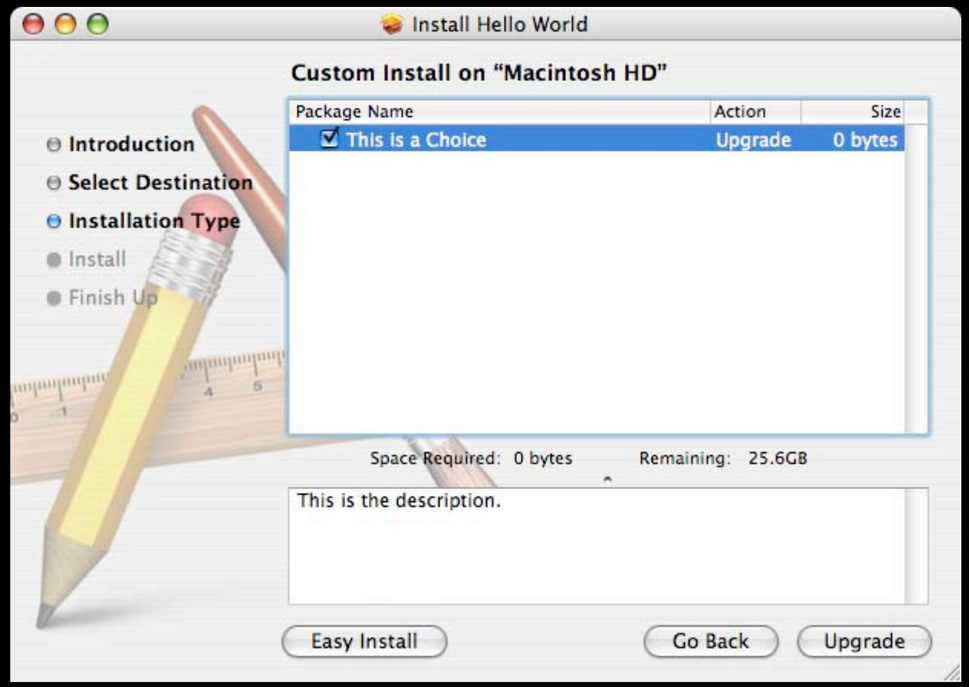
Choices

- Groups packages using <pkg-ref> and <choice> elements
- Selected choices are installed
- Choices can be invisible



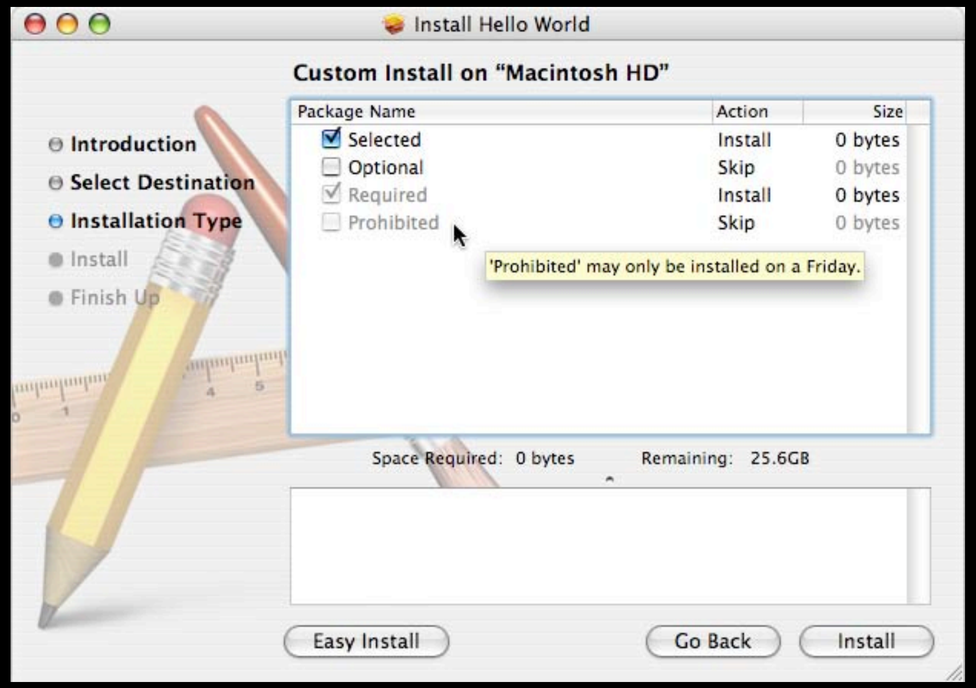
Choices

- Many UI attributes
  - Title
  - Description
  - Tooltip
- These will be initially localized



Choices

- Some attributes are scripted
  - Selected
  - Enabled
  - Visible
- These scripts can be affected by machine state or user input

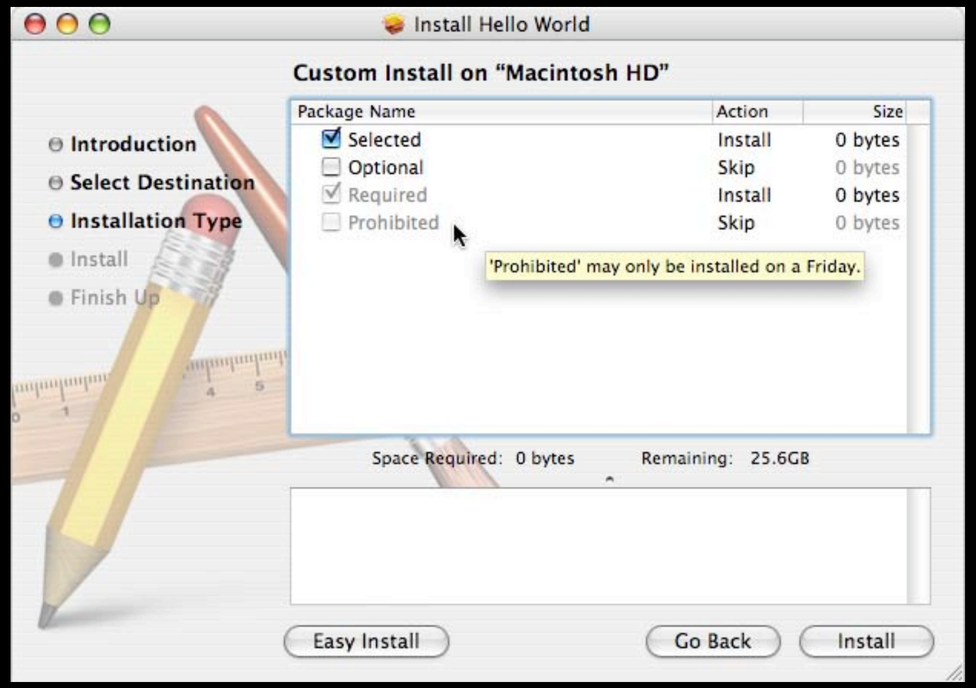


Choices

- Scripted attributes are JavaScript

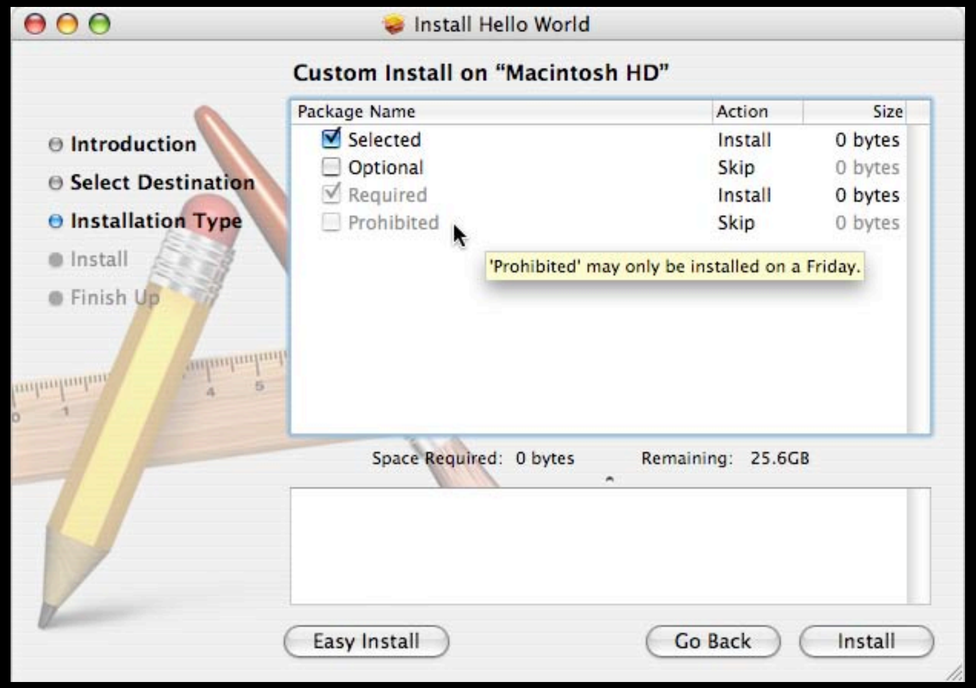
```
<choice
   id='d'
   title='Prohibited'
   selected='false'
   enabled='IsFriday()'
/>

<script>
function IsFriday()
{
   var today = new Date();
   system.log("today is " + today);
   return (today.getDay() == 5);
}
</script>
```



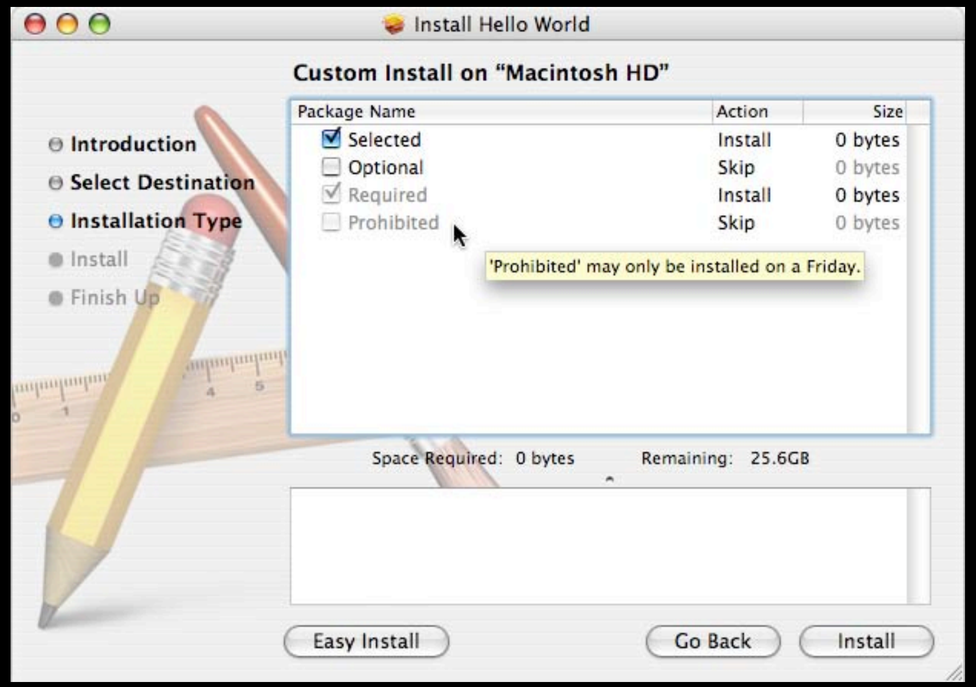
Choices

- Attributes are re-evaluated often!
  - Selected
  - Enabled
  - Visible
- Some are evaluated less often…
  - start_selected
  - start_enabled
  - start_visible
- Switching back to “Easy Install” resets everything and evaluates ‘start_*’ attributes again



Choices

- Attributes can affect each other
- Example: Selecting a plugin might only be allowed if the app is selected or already installed
- With each click, attributes of every OTHER choice are evaluated



Choices

- Evaluating all OTHER attributes prevents most loops
- Since scripts run “one click later”, preserve the user’s intent unless you intend to overrule it

```
<choice id="myFont"
  enabled="choices['myApp'].selected"
  selected="choices['myApp'].selected && my.choice.selected"
/>
```

- Play with three different versions of this in our next Hands On

Hands-On #4

Writing Advanced Choice Attributes with JavaScript

http://developer.apple.com/wwdc2005/

Hands On

Writing advanced choice attributes with JavaScript

- Debug mode gives you more details about how attributes are changing

```
defaults write com.apple.installer DebugChoiceAttrEngine -bool YES
```

- Check out the Installer Log

```
tail -f /var/log/install.log
```

Hands On

Writing advanced choice attributes with JavaScript

- Explore how the three boolean values affect the actions of the choice

```
<choice id="myFont"
  enabled="choices['myApp'].selected"
  selected="choices['myApp'].selected && my.choice.selected"
/>
```

  - If myApp is selected, user's decision stands
  - If myApp not selected, user's decision is overruled and changes are blocked
- Choice attributes can be affected by other choice's attributes
  - Whenever an attribute changes, they all must be reevaluated!
- Gives you control

Hands-On #4

Writing advanced choice attributes with JavaScript

http://developer.apple.com/wwdc2005/

Putting It All Together

Summary

- Packages
- Distribution Scripts
- Requirement Checks
- Choices

What's Next?

- Hands-on samples
- Homework samples
  - Upgrade/downgrade logic
  - Custom locations
  - Multi-disc installs
  - system.run()

What's Next?

- PackageMaker
- Start moving your packages
- Write once, test everywhere
- Explore!

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Mac OS X Lab: Installation and Installer Packages	Tiger Lab Tuesday 5:00PM

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Q & A

Mac OS X Installer Engineering Team

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference